Peer-to-Peer Internet Telephony using SIP

Kundan Singh and Henning Schulzrinne
Department of Computer Science, Columbia University
{kns10,hgs}@cs.columbia.edu

Abstract

P2P systems inherently have high scalability, robustness and fault tolerance because there is no centralized server and the network self-organizes itself. This is achieved at the cost of higher latency for locating the resources of interest in the P2P overlay network. Internet telephony can be viewed as an application of P2P architecture where the participants form a self-organizing P2P overlay network to locate and communicate with other participants. We propose a pure P2P architecture for the Session Initiation Protocol (SIP)-based IP telephony systems. Our P2P-SIP architecture supports basic user registration and call setup as well as advanced services such as offline message delivery, voice/video mails and multi-party conferencing.

1 Introduction

Existing Internet telephony client-server architecture based on IETF's Session Initiation Protocol (SIP [16, 15]) or ITU-T recommendation H.323 [18] typically employ a registration server for every domain. The majority of the system cost is in maintenance and configuration, typically by a dedicated system administrator in the domain. It also means that quickly setting up the system in a small environment (e.g., for emergency communications or at a conference) is not easy. On the other hand, peer-to-peer (P2P) systems [10] are inherently scalable and reliable because of the lack of a single point of failure. P2P systems, in the purest form, have no concept of servers. All participants are peers and communicate in distributed, potentially untrusted environment, to achieve a certain objective such as locating music files or users.

We propose a P2P Internet telephony architecture using SIP. There are two main motivations for P2P-SIP: a fully distributed model to increase robustness, and the ability to deploy without modifying *controlled* infrastructure such as DNS. We analyze various design alternatives and present our P2P-SIP endpoint using Chord [17] as the underlying distributed hash table (DHT). Our novel hybrid architecture allows both traditional SIP telephony as well as user lookup on P2P network if the local domain does not have a SIP server. We use SIP to implement various DHT functions in P2P-SIP such as peer discovery, user registration, node failure detection, user location and call setup by replacing DNS [14] with P2P for the next hop lookup in SIP. To our knowledge, our work is the first such attempt to apply P2P concepts to SIP-based systems.

Besides the P2P scalability and reliability, we have the following additional benefits of P2P-SIP:

No maintenance or configuration: The system works out-of-the-box without requiring any tedious server installation, including NAT and firewall configuration. Our work extends the goals of the IETF Zeroconf [3] Working Group to multimedia communication and collaboration systems.

Interoperability: Unlike other P2P systems such as Skype [2], our architecture uses SIP messages for communicating with other peers. This readily interworks with any existing IP telephony infrastructure such as SIP-PSTN gateways.

These advantages come at the cost of increased *resource lookup delay* and security threats. Unlike $O(1)$ lookup cost in a classical client-server based systems, the P2P lookup cost can be much higher. A distributed P2P architecture makes the system more prone to *security* issues such as trust (privacy and confidentiality) and DoS attacks. A reliable framework for authentication without centralized elements is a challenge. In addition, we may lose some of the traditional IP telephony services. For example, some of the programmable call routing techniques such as SIP-CGI available for SIP telephony can not work in the P2P-SIP system as we do not want to run potentially malicious script uploaded by some peer on our machines.

2 Background and related work

Chord [17] is a ring-based distributed hash table (DHT) for structured P2P systems where each node stores at most $LogN$ entries (or state) in its *finger table* to point to other peers. Lookup is done in $O(LogN)$ time. The *iterative* and *recursive* lookup styles in Chord [17] directly map to the *redirect* and *proxy* behavior, respectively, in SIP. Research in DHT is complementary to our work, since our architecture can use new innovations or optimizations in the underlying DHT.

Skype [2] is a free P2P application based on Kazaa [1] architecture for Internet telephony and instant messaging. The protocol is proprietary unlike SIP. Secondly, it has centralized elements for login authentication [5, 11]. In a way, the Skype architecture is no different from the classical SIP telephony architecture, except that the Global Index Server assigns a *super-node* for a new joining node. The super-node, similar to the SIP registrar, proxy and presence server, maintains the presence information for this node, and locates other users by communicating with other super-nodes. A node that has enough capacity and availability can become a super-node. We believe that the lookup is based on some variation of flooding, similar to Kazaa, instead of using the more efficient DHT-based lookup. The main advantage of Skype is that it implements the equivalent of STUN and TURN servers in the node itself to handle NAT [13], unlike explicit server configuration in existing SIP applications.

Unlike P2P, existing SIP-based telephony [15] has client-server architecture. SIP telephony can be treated as a P2P system with static set of super-nodes (SIP servers) where the lookup is based on DNS instead of a hash key. However, using a pure P2P architecture instead of static set of SIP servers improves the reliability and allows the system to dynamically adapt to node failures.

Table 1 summarizes the similarity and differences between file sharing and multimedia conferencing in the context of P2P. In particular, for Internet conferencing, data storage is not an issue. A single P2P-SIP node can handle many more requests than a file sharing node due to low data volume. Caching of location information is not useful because compared to the file access pattern which follows the zipf distribution, the call access pattern is more uniformly distributed. Moreover, most residential users are likely to get new DHCP IP address every time they connect to the Internet making the cache entry for this user location stale. The file sharing and directory lookup-based systems can tolerate high lookup latency due to the fact that the user does not need to actively wait for the file to download, and the actual file download time tends to be larger than the lookup latency. On the other hand, an IP telephony caller actively waits for the phone on the other side to ring. For file sharing applications, multiple almost-exact copies of a popular file may be available (e.g., independently ripped by different peers). So node reliability does not matter. On the other hand, in the case of IP telephony, we want to talk to the right person, and not some similar person!

Table 1: Different applications of P2P

Properties	File sharing	Conferencing
Data storage	Yes	No
Caching	Yes	No
Delay sensitive	No	Yes
Reliability	Multiple copies	Does not work

3 Design goals

Based on the review of existing P2P systems such as Skype [2] and Chord [17], we propose the following goals for our P2P-SIP telephony architecture.

Zero configuration: The system should be able to automatically configure itself [3], e.g., by detecting NAT and firewall settings, discovering neighboring peers and performing initial registration.

Heterogeneous nodes: It should be able to adapt to available resources and distinguish between peers with different capacity and availability constraints. This favors the distinction between nodes and super-nodes as in Kazaa.

Efficient lookup: Blind search based on flooding is inefficient [6]. The system should use an underlying DHT to optimize lookup. We choose Chord as the underlying DHT for our system because of its robustness and efficiency in the case of concurrent node joins and leaves [9].

Advanced services: It should support advanced telephony services such as offline voice messaging, multi-party conferencing, call transfer and call forwarding as well as advanced Internet services such as presence and instant messaging.

Interoperability: It should easily integrate with existing protocols and IP telephony infrastructure. We choose SIP [15] as the signaling protocol for interoperability.

Besides these explicit goals, there are some implicit scalability and reliability benefits in the P2P-SIP architecture compared to the client-server SIP architecture.

4 Architecture

There can be three designs for using the DHT. On one extreme, the DHT can be used in the server farm among the servers while still maintaining the client-server architecture. On the other extreme, all the nodes become part of DHT. We choose an intermediate design as shown in Fig. 1 where some of the nodes with high capacity (bandwidth, CPU, memory) and availability (uptime, public IP address) are made super-nodes and form the DHT, whereas other ordinary nodes attach to one or more super-nodes without being part of the DHT.

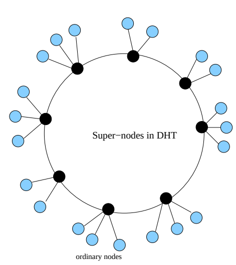

Figure 1: Nodes attached to super-nodes of Chord

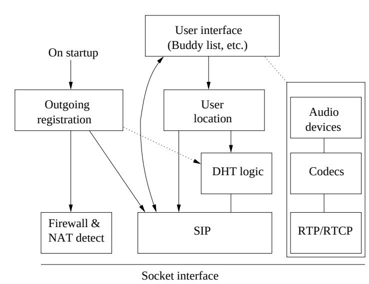

Figure 2: Block diagram of a P2P-SIP node

Fig. 2 shows the proposed block diagram of the different components in our P2P-SIP node. When the node starts up and the user signs-in with her identifier, the registration module is activated to initiate NAT and firewall detection [13], peer discovery and SIP registration. Multicast SIP registration, cached peer addresses from last boot cycle and pre-configured bootstrap addresses are used to discover initial set of nodes. The user interface module keeps track of user's "friends list" and invokes the user location module to locate these friends. User location is obtained using the SIP module or, if this node is a super-node, the DHT module. The DHT module maintains the peer information (e.g., Chord *finger table*) and performs DHT operations such as find, join and leave.

SIP is used as the underlying protocol for locating another user, registering the user, call setup and instant messaging. Once the user location is done, the call setup or instant messages can be sent directly via the SIP module. SIP REGISTER refresh and OPTIONS messages are used to detect node failure. When a super-node shuts down or fails, the registrations are transferred to other super-nodes in the DHT as appropriate. Other SIP functions such as third-party-call control and call-transfer can be implemented in the similar way. The media path (audio device, codecs and transport) is largely independent of the P2P-SIP operation.

Some DHTs (e.g., CAN) may allow parallel search to multiple peers, unlike sequential search of Chord. In this case the super-node may act as a back-to-back user agent (B2BUA) and propagate the SIP message to the neighboring peers. However, parallel search should be avoided to prevent flooding the network, except possibly in the case of emergency call routing, such as 911 calls in the United States.

Scalability: Given that each node in the Chord-based P2P-SIP architecture does $O(LogN)$ registrations, and assuming that each node can support 300 registration requests per second [8] and performs registration refresh every hour, the number of peers, N, can be theoritically $2^{300*3600}$ in a stable P2P-SIP network. Similarly, with a capacity of 90 proxy requests per second [8] and users making one call per hour on an average, the network can support $2^{90*3600}$ users. In practice these numbers will be much less because of other bottlenecks such as bandwidth.

Call Setup Latency: The advantages come at the cost of increased call setup latency. For example, with 10000 nodes in Chord the average lookup path length is six hops [17], so P2P call setup will take about six times more than traditional client-server call setup in SIP. With good network condition, single lookup (INVITE response) in SIP is expected to take less than 200ms. So one or two seconds delay before the phone rings in P2P-SIP is tolerable given that on an average the phone will ring for much longer before the callee picks up.

Some kind of hybrid system may be implemented that takes the advantages of many different structured and unstructured P2P algorithms to further reduce the latency and maintenance cost. For example, there has been recent proposal on one hop lookups for P2P [7] assuming large storage space in the peer nodes.

Security: In addition to authentication and authorization challenges in server-based Internet telephony, we also need to deal with privacy, confidentiality and malicious node behavior in P2P-SIP. For example, a malicious DHT node may not forward the call requests correctly or may log all call requests for future misuse. Hop-by-hop routing of request and responses where each hop (peer) changes the source identifier can be used to provide some confidentiality. Existing P2P reputation systems focus on file sharing (not real-time), have centralized components, assume co-operating peers or have problems of collusion and multiple identities. An electronic credit or debit service is needed to discourage "free riding" [4].

5 Advanced Services

Basic call setup is not enough to be competitive in Internet telephony. This section describes some of the advanced services such as offline message storage and multi-party conferencing for P2P-SIP.

Offline Messages: Existing persistent P2P file storage systems are not sufficient for IP telephony message storage, because IP telephony also needs message waiting indication. We combine storage at sender as well as intermediate DHT node, to provide a more reliable architecture.

Multi-party Conferencing: There are three ways to do conferencing using P2P-SIP. One of the participating members can become the mixer for small scale ad hoc conferencing. Alternatively, a completely decentralized SIP conferencing can be used to establish a full-mesh signaling and media relationship among the participating members. Finally, a multicast media distribution tree can be used assuming a small number of senders at any instant. The tradeoff is in terms of reliability (dependence on single node for mixing), complexity and bandwidth utilization, and requires further study.

6 Conclusions and future work

We propose a pure P2P architecture for SIP telephony. The architecture provides reliability and scalability inherent in P2P systems, in additional to interoperability with existing SIP infrastructure.

We are implementing a P2P-SIP node for multimedia communication using our SIP C++ library. We will be doing performance measurement for reliability and scalability on our actual system instead of using simulations.

More work is needed in advanced services such as large scale application level multicast conferencing using P2P, distributed reputation system for peers, and PSTN interworking related issues such as authentication and accounting. There should be a reasonable incentive to become a super-node to provide services to other peers.

We are working on allowing an internal node inside a firewall and NAT to become a super-node. This reduces the load on public super-nodes, since most of the users typically will be behind some firewall and NAT. Alternatively, the private nodes in a domain can form a secondary P2P overlay connected to the public DHT via a few external connections to reduce the port utilization on the NAT device. Such federation of P2P-SIP networks is for further study.

Some of the open questions described in [12] are relevant to P2P-SIP architecture also. Some kind of hybrid system may be implemented that takes the advantages of many different structured P2P algorithms to further reduce the latency and maintenance cost [7]. Other issues such as regulatory and economic impact, security as well as reliable 911 services are for further study.

Finally, we conclude on a note that unless the SIP servers (proxies, registrars) are widely deployed, we will need P2P based interoperable IP telephony tools so that everyone can use the system. Such P2P-SIP architecture can be extended to other protocols such as H.323, or other DHTs such as Content Addressable Network (CAN).

Acknowledgment

Salman Abdul Baset helped in understanding the Skype architecture. The work is supported by grant from SIPquest Inc., and the National Science Foundation under Grant No. 0202063.

References

[1] Kazaa: peer-to-peer file sharing software application. http://www.kazaa.com.

[2] Skype: Free internet telephony that just works. http://www.skype.com.

[3] Zero configuration networking (zeroconf). http://www.ietf.org/html.charters/zeroconf-charter.html.

[4] E. Adar and B. A. Huberman. Free riding on gnutella. *First Monday*, 5(10), Oct. 2000.

[5] S. Baset. The skype protocol analysis. Project report, Computer Science Department, Columbia University, New York, NY, May 2004. http://www1.cs.columbia.edu/~kns10/research/skype.doc.

[6] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. F. Kurose, and D. Towsley. Modeling peer-peer file sharing systems. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Mar. 2003.

[7] A. Gupta, B. Liskov, and R. Rodrigues. One hop lookups for peer-to-peer overlays. In *HotOS IX: The 9th workshop on hot topics in operating systems*, Lihue, Hawaii, USA, May. USENIX.

[8] J. Lennox. Services for internet telephony. PhD. thesis, Department of Computer Science, Columbia University, New York, New York, Jan. 2004. http://www.cs.columbia.edu/~lennox/thesis.pdf.

[9] D. Liben-Nowell, H. Balakrishnan, and D. Karger. Analysis of the evolution of peer-to-peer systems. In *ACM Conf. on Principles of Distributed Computing (PODC)*, Monterey, CA, USA, July 2002. ACM.

[10] D. Milojicic, V. Kalogeraki, R. M. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. technical report HPL-2002-57 20020315, Technical Publications Department, HP Labs Research Library, Mar. 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html.

[11] A. Rao. Why skype is no different, Mar. 2004. http://gigaom.com/thevoipdaily/archives/2004/03/why_skype_is_no_different.html.

[12] S. Ratnasamy, S. Shenker, and I. Stoica. Routing algorithms for DHTs: some open questions. In *International Workshop on Peer-to-Peer Systems (IPTPS)*, Cambridge, MA, USA, Mar. 2002. IEEE.

[13] J. Rosenberg. Interactive connectivity establishment (ICE): a methodology for nettwork address translator (NAT) traversal for the session initiation protocol (SIP). Internet draft, Internet Engineering Task Force, July 2003. Work in progress.

[14] J. Rosenberg and H. Schulzrinne. Session initiation protocol (SIP): locating SIP servers. RFC 3263, Internet Engineering Task Force, June 2002.

[15] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.

[16] H. Schulzrinne and J. Rosenberg. Internet telephony: Architecture and protocols – an IETF perspective. *Computer Networks and ISDN Systems*, 31(3):237–255, Feb. 1999.

[17] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, San Diego, CA, USA, Aug 2001.

[18] J. Toga and J. Ott. ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations. *Computer Networks and ISDN Systems*, 31(3):205–223, Feb. 1999.